

学習者の編集途中のソースコードに対する 自動フィードバック方法の提案

—処理単位に基づいたフィードバックブロックの定義—

2017SE033 木鎌汐里

指導教員：蜂巢吉成

1 はじめに

大学でのプログラミング教育では、演習中に学生は個別フィードバックを求められることができるが、少人数の教員と数名の TA で行うことが一般であり、大人数の学生に対しての十分な個別フィードバックは実現が難しいという問題点がある。さらに、演習課題に自習で取り組む際にも十分なフィードバックが必要である。演習中に行き詰まり、解けない状態が続くと学習意欲を削がれる原因ともなりうる。既存の自動フィードバックを行うシステムは、正解に近いソースコードが対象であり、編集途中のソースコードに対する自動フィードバックは考えられていない。また、個別フィードバックは、課題を通して学習者に習得させたい知識や学習者の理解度を考慮し、学習者自身が考えながら正解に辿りつくようにする必要がある。

石元 [3] は、ソースコードのバージョン間の差分理解支援ツールを利用したフィードバック方法を提案し、差分理解支援ツールを学習用のフィードバックとして利用するために変数の対応付けと編集スクリプトの改良を行った。本研究ではこの研究を利用してフィードバックの出力部分を実現するが、文単位のフィードバックのみでなく、処理の意味を捉えるようなまとまりでのフィードバックが学習向けには有効である。そこで、次の2つの問題点がある。

問題点 1 教員の負担が少ない方法でソースコードを処理のまとまりに分ける必要がある。

問題点 2 学習者の関心の高い処理のまとまりからフィードバックするように順番を決める必要がある。

本研究では、模範解答と学習者のソースコードを比較し、課題の内容や学習者の理解度に合わせた教育目的の自動フィードバック方法を提案する。C 言語のプログラミング演習中に解法がわからず、行き詰まり状態になる学習者を対象に、間違っている箇所や理解していない文法を把握できるような解法のヒントとなるメッセージを出力する。問題点 1 の解決として、処理のまとまりをフィードバックブロックとして定義し、教員の負担の少ない記述方法を考察する。問題点 2 の解決として、編集途中のフィードバックブロックを学習者の関心のある箇所と捉えることで優先順位を定め、フィードバックを行う順番を決定する。

2 関連研究

Rishabh[1] らの研究では、修正候補を検索し、最小限の修正が必要な修正を決定する MOOC を想定した自動

フィードバックを実現している。

Fangohr[2] らは、テストセットを用意し、学生が提出したソースコードに、教員が予め設定したフィードバックを電子メールで返すことで自動フィードバックを行う。

これらの既存の自動フィードバック方法は、ほぼ完成した状態のソースコードが対象であり、編集途中のソースコードには対応していない。また、そのフィードバック内容は学習者の理解度に合わせた個別フィードバックではなく、一般化した修正方法を提示するものである。

3 学習者に適したフィードバックの分析

学習者に適したフィードバックメッセージについて考える。学習者に適したフィードバックメッセージとは、課題の内容や学習者の理解度、学習者のソースコードの状態を考慮したものである。

ソースコード 1 実数 a の n 乗を求めるプログラム

```
1 double power(double a,int n){
2   int i;
3   double ans;
4
5   ans = 1;
6   for(i = 0; i < n; i++){
7     ans = ans + a;
8   }
9
10  return ans;
11 }
```

まず、学習者の関心によって適切なフィードバックの順番が決まると考える。例えば、Listing1 に対してフィードバックを行う場合、 n を正負で場合分けする if 文からフィードバックするのではなく、for 文の中身である 7 行目の $ans = ans + a$; の式を先にフィードバックする方法がある。このように、記述のない箇所よりも学習者が取り組んでいる箇所を先にフィードバックする方がわかりやすいと考える。さらに、7 行目の $ans = ans + a$; の式は、5 行目の $ans = 1$; で初期化し、6 行目の for 文で n 回繰り返すことで a の n 乗の計算式として成り立つので、この 1 文だけではなく、5 行目から 8 行目までを a の n 乗を計算する箇所として、まとめてフィードバックする方が望ましい。以上のことを踏まえると、上から順番に文単位のフィードバックを返すよりも処理のまとまりごとに、学習者が関心の高い箇所からフィードバックすることで、プログラムの全体像を捉えやすくなり、学習者自身に考えさせるフィードバックとなる。何を行う箇所なのかを説明した後、その処理に必要な文単位でのフィードバックを出力する。

学習者の習熟度を考慮すると文単位のフィードバックメッセージの内容は複数通りあると考えられ、河本 [4] の研究で、文単位のフィードバックメッセージ生成方法を提案している。

本研究では、学習者の編集状況を把握し、フィードバックを行う順番を決定する方法を提案する。

4 フィードバックブロックの定義

学習者のソースコードから取り組んでいる箇所を特定し、適切なフィードバックの順番を決定するためにソースコードを処理のまとまりで区切る。この処理のまとまりをフィードバックブロック（以下、FB ブロック）と呼ぶこととする。

FB ブロックの分け方は一つに定まるとは限らず、自動で定義ができないので、教員がソースコードに合わせて処理のまとまりとなるよう定義する必要がある。関数内の全てのソースコードは文単位でそれぞれ必ず FB ブロックに属するものとし、以下を目安として定義すると良い。

- 制御文は「条件分岐」や「反復」などの処理を行う文であり、制御文を FB ブロック分けの基準とする。
- 初期化の式はデータフローで繋がる FB ブロックに属すると考える。
- 変数宣言は変数宣言で 1 つの FB ブロックとする。

FB ブロックは予め教員側が模範解答に XML で記述する。タグ付けは制御文を目安に自動で行い、教員がソースコードに合わせて適切な FB ブロックとなるように調整して、属性に FB ブロックの説明を付加する。

教員が記述した FB ブロックを分類し、学習者の関心の高い順番となるよう優先順位を次のように定める。

1. 間違いを含む FB ブロック
2. 完成した FB ブロックを子に持つ FB ブロック
3. 記述のない FB ブロック
4. 未完成の変数宣言の FB ブロック

Listing1 をこの優先順位でフィードバックの順番を決定すると、FB ブロックに関して出力されるメッセージは次のようになる

1. 実数 a の n 乗を計算する箇所をフィードバックします。
2. n が正の場合を記述しましょう。
3. n が負の場合を記述しましょう。
4. n が負の場合の実数 a の n 乗を計算する箇所を記述しましょう。

5 自動フィードバックシステムの提案

模範解答と学習者の編集途中のソースコードを比較し、学習者のソースコードの状態と課題の内容を考慮し、学習者の理解度に合わせたフィードバックメッセージを出力する自動フィードバックシステムを提案する。

模範解答に FB ブロックを記述したものを用意する。記述された FB ブロックから、模範解答と学習者の AST をそれぞれ FB ブロックに分ける。石元 [3] の研究で学習用に改良された編集スクリプトを元に、FB ブロックを分類する。4 で定めたフィードバックの優先順位で分類された FB ブロックのフィードバックを行う順番を決定し、フィードバックメッセージを出力する。

6 考察

提案したシステムを、記述した通りの FB ブロックに分けられるか、定めた優先順位の順番でフィードバックメッセージが出力されるかの 2 点より評価した。制御文を複数含む関数を対象とした。前者の結果は、7 つのソースコードのうち 6 つで正しく FB ブロックに分けられた。後者の結果は、間違いを含んだ 39 のソースコードのうち、33 のソースコードで期待通りのフィードバックメッセージが得られた。期待通りの結果が得られなかった原因は、編集スクリプトに冗長な操作が含まれていたことであった。したがって、本システムによって対象とした 8 割のソースコードで期待通りの結果が得られ、制御文の入れ子を含むソースコードでも XML で正しく FB ブロックを定義し、フィードバックの順番を決定できる。

7 おわりに

本研究では、学習者や課題に合わせた教育目的のフィードバックを目的に、ソースコードを意味のまとまりで分けてフィードバック箇所を特定し、学習者のソースコードの状態に合わせた順番でメッセージを出力する自動フィードバック方法を提案した。今後の課題として、優先順位の例外への対応、実際の教育現場での検証、文単位のフィードバックも踏まえた提案手法全体の自動化が挙げられる。

参考文献

- [1] Rishabh Singh, Sumit Gulwani, Armando Solar-Lezama: "Automated Feedback Generation for Introductory Programming Assignments", ACM SIGPLAN Conference on Programming Language Design and Implementation, pp.15-26(2013).
- [2] Hans Fangohr, Neil O'Brien, Anil Prabhakar, Arti Kashyap: "Teaching Python programming with automatic assessment and feedback", arXiv:1509.03556[cs.CY](2015).
- [3] 石元慎太郎: "プログラミング学習者の編集途中のソースコードに対するフィードバック方法の提案", 南山大学理工学部 2019 年度修士論文 (2019).
- [4] 河本葉々: "プログラミング学習におけるソースコードの文に対するフィードバックメッセージ生成方法の提案", 南山大学理工学部 2020 年度卒業論文 (2021).